ERDC/ITL SR-01-1

Information Technology Laboratory

US Army Corps
of Engineers®
Engineer Research and
Development Center

Preliminary Investigation of the Visualization of Buildings in the OneSAF Environment

Michael E. Pace and Warren R. Bennett

September 2001

ERDC/ITL SR-01-1
September 2001

Preliminary Investigation of the Visualization of Buildings in the OneSAF Environment

by Michael E. Pace, Warren R. Bennett

Information Technology Laboratory
U.S. Army Engineer Research and Development Center
3909 Halls Ferry Road
Vicksburg, MS 39180-6199

Final report

Approved for public release; distribution is unlimited

Prepared for U.S. Army Corps of Engineers
Washington, DC 20314-1000

Contents

List of Figures

Preface

This report describes the investigation of the visualization of buildings in the One Semi-Automated Force (OneSAF) environment which is a new Army simulation program capable of modeling entities within buildings. The aim was to better support the modeling of buildings for use in Special Forces Operations. This study was performed with funds provided by the Directorate of Military Programs, Headquarters, U.S. Army Corps of Engineers, under the Research, Development, Test, and Evaluation Program.

This report was written by Mr. Michael E. Pace, Computer-Aided Engineering Division (CAED), Information Technology Laboratory (ITL), U.S. Army Engineer Research and Development Center (ERDC), Vicksburg, MS, and Mr. Warren R. Bennett, Computer Science Division, ITL, ERDC.

The work was performed under the general supervision of Mr. H. Wayne Jones, Chief, CAED, ITL, ERDC, and Mr. Timothy D. Ables, Acting Director, ITL.

At the time of publication of this report, Dr. James R. Houston was Director of ERDC, and COL John W. Morris III, EN, was Commander and Executive Director.

1 Introduction

Background

The next generation of Models and Simulation (M&S) for use in determining future force design, developing doctrine, training military leaders, and performing acquisition for the Army After Next is currently under development. The Army M&S program is developing software called One Semi-Automated Force (OneSAF) to fulfill the objectives of M&S. Special Forces units train for terrorist scenarios, such as capture and occupation of structures that may include innocent hostages. Given available information about the building at the time, a realistic simulation is desired in order to train to neutralize the threat. At present, OneSAF has a capability to perform military operations within buildings, but a rapid method to produce a building model for the synthetic environment does not exist.

In September 1999, the Information Technology Laboratory (ITL), U.S. Army Engineer Research and Development Center (ERDC), Vicksburg, MS, began an investigation of the visualization of buildings in the OneSAF environment. This report details the findings of this preliminary study.

Purpose of Study

This effort is an attempt to apply M&S tools to the specific problem of visualizing structure interiors for Special Forces operations. These operations are rapid response operations that must be conducted within hours of the initial invasion of the structure. A target deadline of 3 days has been given as the metric for performing the modeling and training for the operation. Since 3 days is the deadline for planning the operation, the visualization must be ready as soon as possible for the planning to begin. Our target is to have the visualization available in 36 hr after receipt of structure data.

The study will focus on the identification of the modeling processes and technical challenges necessary to incorporate buildings into the OneSAF environment. To identify these processes and technical challenges, several questions must be answered:

a. What are the generic stages involved in the process of modeling a building?

b. How are buildings represented geometrically in the OneSAF environment?

c. How do buildings interact with entities in the OneSAF environment?

d. What is the current procedure for creating a building?

e. What are the current software packages used to model the geometry of buildings?

f. How are buildings inserted into the database used by OneSAF?

g. What other commercial off-the-shelf (COTS) and Government off-the-shelf (GOTS) graphics packages are available to aid in the modeling process?

Answering these questions will provide a foundation for further study and basis for improving the present system.

The ultimate goal of future work would be to develop a capability to rapidly generate the external/internal spatial characteristics of a multistory building from multiple input sources (blue-line paper, Computer-Aided Drafting and Design (CADD) drawing, special-purpose modelers) with levels of detail dictated by the functional requirements of the required training mission within a 36-hr time period. Associating the building model with the structural attributes required to perform an engineering analysis of the building is also desirable. This would produce a rapid structural modeling system capable of creating building-specific input for synthetic environments with the capability to perform a physics-based analysis.

Organization of Report

This report is organized into chapters that address the state of modeling and capabilities that presently exist pertaining to buildings. A discussion of general modeling processes to proceed from a paper format to the format required for the OneSAF program are presented. Finally, various software packages, file formats, and programming controls and Application Programmer Interfaces (APIs) are discussed that might shorten the modeling time required. Appendix A gives the reader definitions and common graphics terms which might be encountered in the literature. Appendix B lists software discussed in the report with prices.

2 Simulation Background

OneSAF Overview

The OneSAF program is the next generation of simulation programs being developed by the Army. A brief description of the program follows.

Background

In January 1996, the Computer Generated Forces (CGF) assessment working group recommended that a flexible, composable, SAF architecture be developed that could integrate the best features of the Modular Semi-Automated Forces (ModSAF) and the Close Combat Tactical Trainer (CCTT) SAF. In May 1997, the Deputy Commanding General (DCG), TRADOC, approved the Mission Need Statement (MNS) for the OneSAF model. The Deputy Undersecretary of the Army (Operations Research) (DUSA(OR)) then directed that a cross-domain Integrated Concept Team (ICT) be chartered that would be tasked with the development of a requirements document, acquisition strategy, and program management plan. The DUSA (OR) asked the Army Materiel Systems Analysis Activity (AMSAA) to lead the effort (OneSAF ORD 1999).

OneSAF is designed to be a composable CGF that can represent a full range of operations, system, and control processes from the individual combatant and platform level to battalion level. OneSAF will have a variable level of fidelity that supports all M&S domains and employs appropriate representations of the physical environment and its effect on simulated activities and behaviors.

OneSAF should support special forces operations in natural or Military Operations in Urban Terrain (MOUT). Therefore, OneSAF should interact with natural and man-made features such as buildings with usable exterior and interiors. The battlefield and supporting infrastructure should be able to be modified with respect to mobility, countermobility, survivability, and sustainment engineering. Objects such as buildings or bridges could be destroyed by munitions.

Operation

The visual display of a running simulation can either be viewed on a plan view display (PVD) or in a stealth view. The PVD view is essentially a two-dimensional (2-D) representation of the three-dimensional (3-D) terrain data. The stealth view is a 3-D view of the synthetic environment.

The front-end workstation for OneSAF is called the SAFstation and provides human supervisory control of the OneSAF entities through the use of a graphical user interface. The user can load scenarios, create battlefield entities, assign tasks to friendly/enemy vehicles, influence their behaviors, or simply examine the simulated battlefield on the PVD (Lockheed Martin Advanced Distributed Simulation (LMADS) 1999a, 1999b).

The stealth viewer is a special application that allows the user to monitor battlefield events from any perspective. The viewpoint can be attached to an entity or a fixed distance from an entity. The viewpoint can be switched on the fly to view the battlefield from any entity's perspective (Lockheed Martin Informaiton Systems (LMIS) 1999). A GOTS package called ModStealth can be used to visualize the simulation in 3-D. A COTS package called MakStealth can also be used to view the 3-D visual database.

Representation of Buildings in OneSAF

The representation of buildings in OneSAF can range from very basic to complex. A basic building could be represented as a volume with texture maps on the exterior to provide a visual appearance of a building with windows, doors, and architectural detailing. The building is essentially an overturned box on the terrain that affects the mobility and intervisibility of objects.

A building could also be represented as a multielevation surface (MES) structure. This type of structure is described in more detail later in this chapter. Essentially, the exterior (doors, windows) and interior (walls, ceilings, floors, doors) of the building are modeled in a certain format to allow movement through the building.

Buildings can also exist as multistate objects (MSO). This allows the buildings to sustain damage from munitions and have their visual appearance change accordingly. The computation of damage can be performed through look-up tables or keyed to results from physics-based analyses (Birkel and Lukes 1998, Chapter 3).

MES Structures

The information in this section is taken from Birkel and Lukes (1998, Chapter 4), Stanzione et al. (1996a, 1996b), and the Distributed Warrior

Network (DWN) MES Creator Developer's Guide (Science Applications International Corporation (SAIC) 1998) and describes the details of an MES structure.

Background

The Improved Computer Generated Forces Terrain Database (ICTDB) and Dismounted Infantry Semi-Automated Forces (DI SAF) projects resulted in additional capabilities in the ModSAF program that later transitioned to become the basis for the OneSAF program. The ICTB project added the concept of a MES structure to the synthetic environment. An MES structure can be a bridge, building, cave, or tunnel. The MES structure enabled an entity to navigate under (e.g., a bridge) or inside (e.g., a building) the structure. The DI SAF project added the capability for dismounted infantry (soldiers) to move around in and interact with the synthetic environment. The soldiers were given the ability to operate within the MES buildings.

The ICTDB project started in August 1994, with development completed in September 1997. ICTDB was one of four Defense Advanced Research Projects Agency (DARPA) Synthetic Environment (SE) programs, which included the Total Atmosphere and Ocean Services (TAOS), Dynamic Virtual Worlds (DVW), and Dynamic Terrain and Objects (DTO) projects. The main goal of the ICTDB project was to extend the terrain database representation for the DARPA Synthetic Theater of War (STOW) Joint Semi-Automated Forces (JointSAF) Computer Generated Forces (CGF) program. ICTDB was a joint project of The Analytical Sciences Corporation (TASC) in Reading, MA, and SAIC in Burlington, MA (Birkel and Lukes 1998, Chapter 4).

The DI SAF program was started as a part of the Distributed Warrior Network (DWN), under the Advanced Distributed Simulation Technology II (ADST II) program, in June 1996. The contract was awarded by Simulation, Training and Instrumentation Command (STRICOM) and the work was performed by the Asset Group from SAIC. A second phase of work on DI SAF began in September 1997 as part of the DWN Enhancements for Restricted Terrain Delivery Order and focused on adding behaviors for urban terrain to the DI SAF program (Reece et al. 1998).

Geometrical representation of MES structures in the Compact Terrain Database (CTDB)

MES structures are a new volume feature in the CTDB. The feature represents MES structures as both abstract features and full 3-D models. MES structures are represented with enclosures and apertures. The enclosures are defined by solid surfaces such as walls, floors, and ceilings. An enclosure is the room, hallway, or interior portion of a building that allows movement. Apertures are the connecting areas between enclosures such as doors and windows. Apertures allow the movement from one enclosure to

another. Topological information between enclosures (via apertures) is provided for movement and planning purposes.

The new volume feature consists of a roof outline and a reference to an MES structure. The MES structure consists of a header, a list of enclosures, a list of apertures, a list of triangles, and a hierarchy of grid boxes.

The header specifies the origin of a local coordinate system for the MES in the Global Coordinate System (GCS). The origin can be chosen to be in any relation to the MES. The header also specifies a transformation matrix from the world coordinates to the MES local coordinates. This allows the MES structure to be positioned on the terrain and allows reuse and relocation of the MES. The header also specifies a fixed point basis (meters/unit) that is the unit in which all MES coordinates are stored.

Enclosures are an enclosed volume that can be concave or convex. Each enclosure must be a single level and cannot overlap or be contained within another enclosure. The only exception to this is the exterior enclosure that contains all other enclosures. There can be only one positive normal surface (a floor) and one negative normal surface (a ceiling) for each enclosure. Enclosures can be arbitrarily oriented in relation to each other and are usually grouped by story. The enclosure data also contain a floor outline and the average height of the enclosure. Enclosures block visibility and mobility. See Figure 1 for a typical MES building.

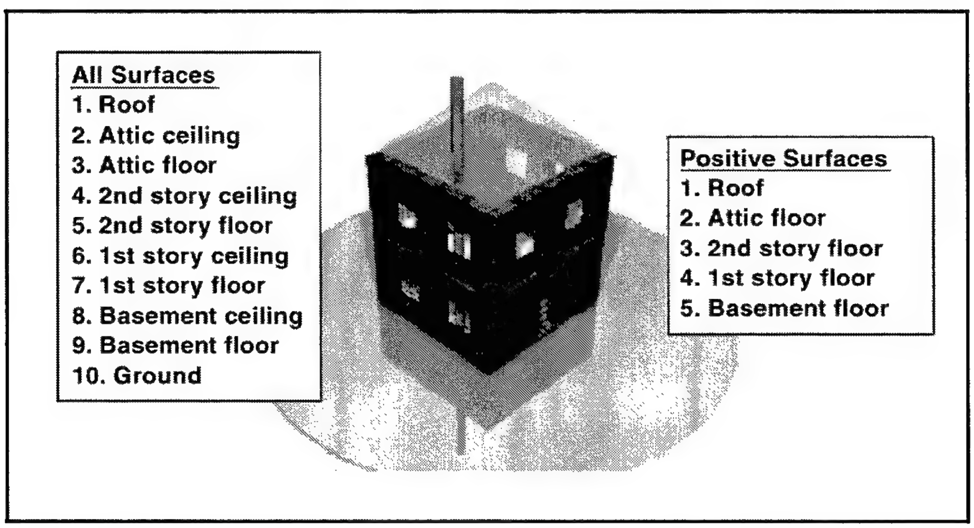

Figure 1. Typical MES building (from Birkel and Lukes 1998)

The walls of an enclosure are composed of triangles. Enclosure triangles must not intersect, be shared, or be overlayed (coplanar) with triangles from any other enclosures. Any edges of enclosure faces must share vertices at the endpoints of the edges, i.e., there can be no T-vertices. Each enclosure contains a list of wall triangles and apertures. The apertures are also composed of triangles. The union of triangles for the enclosure and apertures must form a closed volume. That is, there must not be a path

from the interior to the exterior of the enclosure that does not cross a triangle. Enclosure triangles are always defined such that their front face is defined by a counter-clockwise winding of the vertices of the face. The front face of enclosure triangles always points **inward** to the enclosure. The triangles for the exterior enclosure should be oriented such that their front faces point **outward** for the MES. The exterior enclosure is the exterior of the MES and only contains apertures that connect it to enclosures on the inside of the structure.

Two adjacent enclosures are connected by one or more apertures. Apertures must be planar and connect exactly two enclosures. Apertures are an N-sided polygon outline at the juncture of two enclosures. The aperture is represented as the triangulation of the outline. An aperture has the following attributes:

Attribute	Description
Material	Material type of the aperture
Wall thickness	(2) entries - thickness of wall on either side of the aperture
Mobility	Boolean 0/1 indicating if aperture is penetrable
Step to enclosure	Used to determine if a unit can traverse the aperture
Visibility	Range 0.0-1.0 indicating opacity of aperture (0 = totally opaque)

Apertures should be created right at the edge of where one enclosure butts against another, as opposed to in the center of a doorway or window. In this model, that would mean that one of the wall thickness parameters is always 0, and the other is the thickness of the door or window frame.

Interaction of MES Structure with the Environment

The PVD shows the interior of MES buildings (Figure 2). This display allows the placement of individual combatants (IC's), routes, and objects inside the building. The display also allows the viewing of objects inside the building. The display shows interior walls and marks apertures as doors or windows and open or shut (Reece et al. 1998).

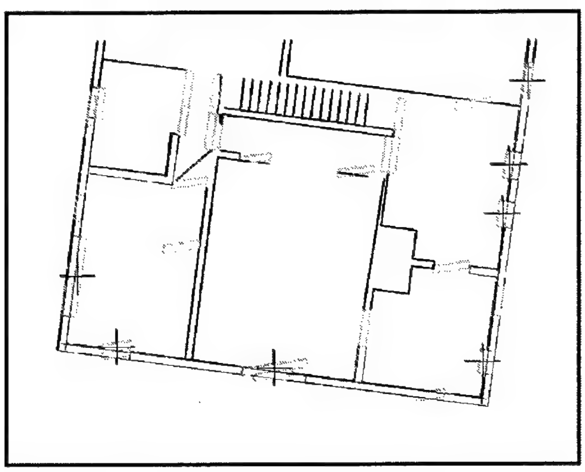

Figure 2. PVD showing MES interior

As part of the DWN project, dismounted infantry were given the capability to move within an MES building. Building clearing operations with IC's are possible. These operations include blowing a hole in a wall, entering the building, and engaging the enemy.

A method was added to ModSAF to allow dynamic hole creation. Methods were added to allow blowing a hole in a wall or shooting out a door. Each simulator node creates a hole or opens a door in response to the receipt of a detonation protocol data unit (PDU) on the distributed interactive simulation (DIS). Each system uses the same repolygonalization algorithm to create the same hole. There is no need for a server to send terrain change notices or for the systems to process them. An AT-8 round creates a large building hit, an SAW creates a medium hit, and an M16A2 produces a small hit. A large hit produces a hole and changes the polygonal geometry and topology of the MES, a medium hit only opens apertures, and a small hit has no effect (Reece and Dumanoir 1998).

Present Procedure for Modeling Buildings

Two different databases are needed for the OneSAF program. A Compact Terrain Database (CTDB) is needed for the 2-D display and control of the simulation, and a 3-D visual database is needed for the ModStealth or MakStealth viewers. At the present, the creation of these databases containing an MES building must be performed using several tools.

MES buildings are labor intensive to generate by hand and subject to human error. To alleviate some of this difficulty, SAIC developed the MES Toolkit to aid in the modeling of an MES building. The toolkit is a plug-in designed to operate within AutoCAD and is discussed further in Chapter 4.

The toolkit uses an OpenFlight file as a template to aid in constructing the MES building. The OpenFlight file is produced by the Multigen modeling and simulation program and contains the 3-D geometry of the building. The building is modeled in a specific manner to define enclosures (rooms) and apertures (doorways, windows). From this information, the MES Toolkit produces a reader file (.rdr extension), also called a correction file that contains the MES information. This file can be compiled into the CTDB terrain database by using the recompile program. The same building can be placed multiple times on the terrain at different orientations using the reader file.

To produce the 3-D visual database, the terrain generation toolkit called S1000 Toolkit from the U.S. Army Topographic Engineering Center (TEC) can be used. The toolkit has a modeling module that can be used to produce 3-D geometry. The S1000 source data can be compiled into the OpenScene format used by the ModStealth program. The S1000 Toolkit cannot directly produce a CTDB database that contains an MES building.

Using Reader Files to Add Buildings to a Terrain Database

The CTDB recompile program can use a reader file to insert a building into the CTDB database. Since MES buildings are not supported by the S1000 toolkit, a quick method to insert an MES structure into the CTDB is to use a reader file (Birkel and Lukes 1998, Chapter 4, Stanzione et al. 1996a, 1996b). The ADD_MES_TEMPLATE operator allows an MES structure to be defined, and the ADD_MES_VOLUME operator allows the instantiation of the structure in the database. When one of these operators is encountered in the correction file, the compiler creates a data structure for an MES template or volume as appropriate and passes it to the compiler back end for processing. In the correction file, each MES volume is linked to its template according to a template name chosen by the human modeler. The following format is used to add a new MES volume, such as a building, a bridge, or a tunnel. <template-name> is the link between the new volume and its corresponding template:

```
(ADD_MES_VOLUME <template-name>
(origin <real x><real y><real-z>)
(cell <integer cell-id>)      ;; GCS, required even in non GCS mode
(mes_to_world_rotation_matrix  ;;right-handed coordinate system Y up
<real><real><real>
<real><real><real
<real><real><real))
```

The following format is used in the correction file to add a new MES template:

```
(ADD_MES_TEMPLATE <template-name> (type <type>)
(meters_per_unit <real>)
(roofline <vertex><vertex><vertex>...)
(floorline <vertex><vertex><vertex>...)
(enclosures <enclosure><enclosure>...)
(apertures <aperture><aperture>...))
```

where <enclosure> is:

```
(<enclosure-name>
(average_height <real>);;meters
(footprint <vertex><vertex><vertex>...)
(triangles <triangle><triangle><triangle>...))
```

and <aperture> is:

```
(<aperture-name>
(visibility <real>) ;; 0 <= visibility <= 1
(mobility <real>) ;; 0 <= mobility <= 1
(connects_to_<connection><connection>) ;; exactly two connections
(outline<vertex>,vertex>,vertex>...)
(triangles<triangle><triangle>...))
```

<triangle> is defined as:

```
((material <integer>) ;; FACC MCC
(vertices <vertex><vertex><vertex>)) ;; exactly three vertices
```

with <vertex>:

```
(<real x><real y><real z>) ;; local MES coordinates
```

<connection> is defined as:

```
(<enclosure_name>
(step_to_enclosure <True/False>)
(wall_thickness <real>))
```

and <type> is one of:

```
unknown
building
cave
tunnel
bridge
other
```

<template-name>, <enclosure-name>, and <aperture-name> are arbitrary, nonquoted strings.

The following example demonstrates how to compile a reader file called my_mes.rdr into a database called dwn.c7b to create an output CTDB (including an MES) called dwn-mes.c7b (line breaks have been added for clarity):

```
./recompile
no_mesrel
datapath /proj/ModSAF/common/data
input_ctdb dwn.c7b
tdbpath /tmp
output_base_name dwn-mes
correction_files my_mes.rdr
```

This process will run for several minutes and will report the insertion of the MES volume and template structures into the database. Following is a brief description of the parameters used:

`-no_mesrel`	Use the Z value of the origin of the MES as an absolute Z value. By default, the Z value in the origin is an offset from the terrain underlying the model's origin
`-datapath`	Specify where to find ModSAF data files (`coord.rdr`)
`-input_ctdb`	The FULL pathname to the input database into which the MES will be inserted
`-tdbpath`	Path to where the output database should be created
`-output_base_name`	The file name portion of the output database ONLY, not including any directory paths or the extension
`-correction_file`	The `.rdr` file that contains the MES volume and template created by AutoCAD and hand-editing

3 Examination of Modeling Process

Overview of Processes to Model a Building

A broad conceptual overview of the modeling and interaction of buildings in the OneSAF synthetic environment is shown in Figure 3. This figure depicts the modeling of a building using various COTS and GOTS packages. Additional specialized modelers add structural attributes. The visual and engineering models are combined to provide information to OneSAF that allows building interaction with the environment based on physics.

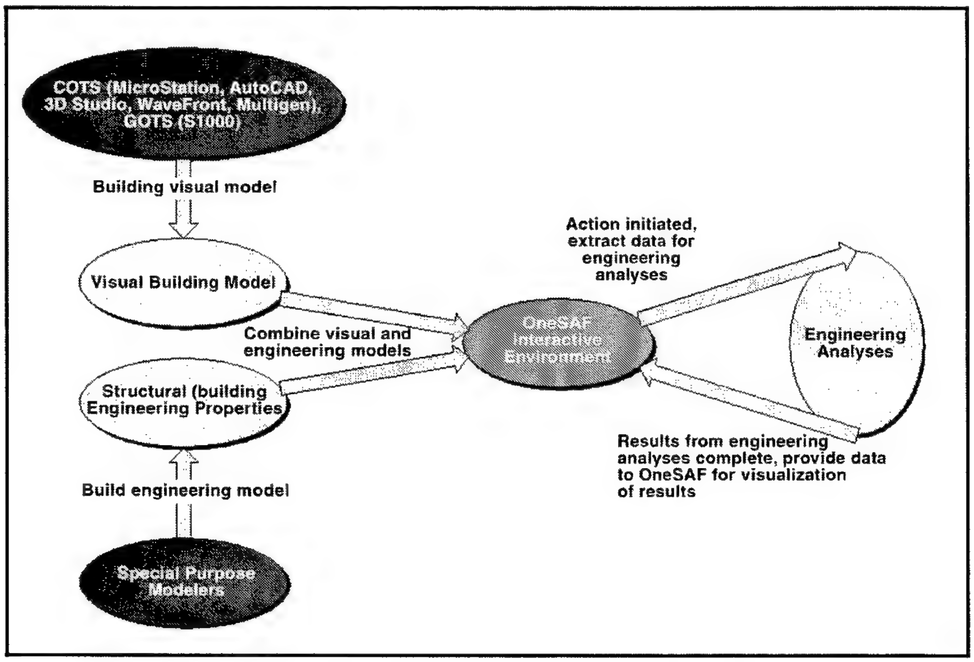

Figure 3. Overview of building interaction with OneSAF

The process of producing a building model is shown in Figure 4. The process consists of six main stages. The information for a building starts out as a line drawing on a blueprint. This information is used to produce a

2-D file format. This 2-D file includes information about the wall, door, and window locations. The 2-D file is manipulated to produce a 3-D file format containing the geometry of the building. This 3-D format includes the height of the walls. Textures can be added at this stage or later in the process. A data extraction process produces a neutral file format that can be used to produce both 2-D and 3-D views of the building. These data can be converted for use in the S1000 Toolkit, MES Toolkit, or other software yet to be developed. These programs would produce the files necessary for input into the terrain compilers. Finally, the terrain compilers would produce the CTDB and 3-D visual (GDE) databases necessary for use in OneSAF and ModStealth.

The processes that transform or carry data from one modeling stage to the next may accept additional data at each stage or output data that can be used in another stage. These processes will involve the use of COTS and/or GOTS packages to perform the data manipulations.

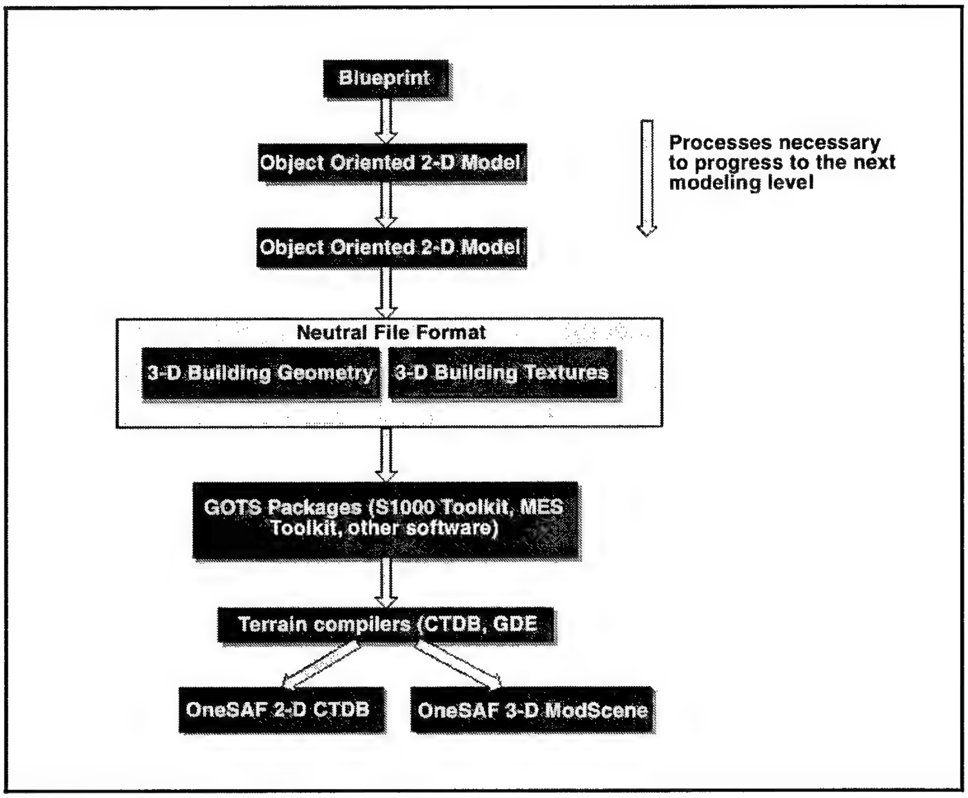

Figure 4. Flowchart of modeling process for a building

Description of Processes to Model a Building

Original data forms

Data available to model a building can range from a hand-drawn sketch, blueprint of floor plans, architectural elevations, 2-D CADD drawings, or

3-D CADD drawings. Each of these starting points represents an entry into the process depicted in Figure 4.

In the simplest format, these data for a building are contained in a paper drawing. This drawing could contain information such as the 2-D layout of the building showing locations of walls, doors, and windows. Other miscellaneous information, such as the location of electrical fixtures, power supplies, plumbing locations, and heating, ventilation, and air condition (HVAC) systems, could also be shown. More modeling time will be required the further the starting point is to the top of the process diagram. That is, modeling from a blueprint will require more time than modeling from a 3-D CADD file.

Transformation to 2-D data

The transformation from the paper copy to an electronic file format can be accomplished in several ways. The blueprint can be scanned to produce a raster image of a particular floor. This raster image can be converted to a vector image by using raster-to-vector conversion software. There is always some kind of cleanup needed after this type of conversion. Items such as dimension lines, text, and architectural symbols must be deleted from the drawing. Only the geometry of the floor plan is needed. After cleanup, a vector image would exist that describes the 2-D layout of the particular floor.

Another method would be to hand digitize the floor layout on the blueprint using a digitizing tablet. This requires intensive human interaction. The graphics operator can decide at the time of digitizing what is important to digitize and what is not. The blueprint could also be scanned in to provide a template for heads-up digitizing of the floor plan. Finally, the blueprint could be used just for reference in redrafting the floor plan from scratch.

These data that are generated at this stage must be captured in some format. One file format that is very popular among COTS and GOTS packages is the DXF format. The details of this format vary from version to version of software packages but is still widely supported as a means of transferring geometry information.

Transformation to 3-D data

The conversion from 2-D to 3-D data involves some knowledge of the floor heights and locations of openings such as windows. Interior and exterior doors usually have standard dimensions that may aid in their placement. Windows will require information such as the sill height, window height, and window shape to be placed correctly in the building model.

The process of converting a 2-D format to a 3-D format could involve either the process of extrusion or an intelligent 2-D modeling capability. Software packages are available that can extrude 2-D lines to produce 3-D surfaces. The extrusion process would raise the walls perpendicular to the floor plan by the height of the building story. After the extrusion process, windows and doors would need to be placed in the model at their appropriate location and size. The openings would then be placed in 3-D. Some software packages possess intelligent objects that can position themselves correctly on a wall. The packages allow the user to draw in 2-D while producing a 3-D drawing with objects correctly positioned.

After the 3-D geometry is modeled, textural information could be added to the building to make it look more realistic.

Transformation to a neutral file format

As mentioned earlier, the DXF file format can store 3-D geometrical information and is supported by many of the COTS and GOTS packages. The disadvantage of the DXF format is that textures are not supported. Therefore, at this stage of the modeling process, geometrical and textural information may need to be translated into another file format.

Transformation to CTDB and OpenScene files

The main tools available to produce the CTDB and 3-D visual (OpenScene) databases for use by OneSAF and ModStealth are the S1000 Toolkit and the MES Toolkit.

The main tool that can be used to produce the 3-D visual (OpenScene) database is the S1000 Toolkit. The geometry and textural data will need to be in a format that the S1000 Toolkit can use. At the present, S1000 can import AutoCAD DXF files to produce the 3-D geometry. Textures can be applied within S1000. Finally, S1000 can be used to produce the OpenScene database. S1000 cannot directly produce a CTDB database containing an MES structure.

The MES Toolkit is used to model buildings in the required format needed to produce a CTDB terrain database. The MES Toolkit operates inside AutoCAD, and DXF files can be imported to aid in the definition of the geometry of the building. Other formats such as OpenFlight files can also be imported and used to construct the building.

Other methods to produce MES buildings in the CTDB or OpenScene format will require development of specialized software.

Information Required for a Walkthrough

Visualization of the interior of buildings can be done to various degrees. It is important to note that there will be a tradeoff between the level of detail modeled and the responsiveness of the simulation. A very basic building simulation could involve enough detail to show the general shape of the exterior of the building, the openings (windows and doorways) and the interior walls, floors, and ceilings. This would provide enough information to perform a basic walkthrough to familiarize a person with the layout of the building. More detail such as doors, doorknobs, light switches, lights, electrical systems, HVAC systems, sewer lines, and furniture would add more realism to the simulation. Textures would add more detail to the model by helping to visualize the actual floor and wall finish treatments (e.g., paint, wood flooring, carpet, wall paneling, etc.). The level of detail modeled will be dictated to some extent by the training requirement and available information.

Some of this information (such as location of electrical lines, light switches, and door orientation) could be very difficult, if not impossible, to know. As-built drawings may be nonexistent. Even if as-builts are available, some information such as electrical wiring placement or electrical outlet locations may not be shown.

3-D Visualization Overview

Modeling can be performed for several different purposes. Each purpose will require information about geometry, visual appearance, or behavior and interaction within the virtual environment.

Modeling for various purposes

Modeling is performed to facilitate (Costello and Bee 1997):

a. Product design.

b. Presentation.

c. Real-time simulation.

Traditional CADD-type packages are used to support analysis and design of engineered items such as buildings, mechanical parts, structural components, consumer products, and civil projects like locks and dams. The CADD packages are used to accurately model the geometry of an object for subsequent analysis and design. The modeling can be used to produce finite element meshes for complex engineering analyses, the results of which can be used to provide the most economical design. Products such as AutoCAD and Microstation fall into this category. These packages

are not necessarily designed to work in real-time. Their focus is on accurate modeling of the geometry of the object.

The purpose of some modeling is to produce presentation material to facilitate understanding of a conceptual design, to illustrate a procedure, or for marketing purposes. This type of modeling will include features for photorealistic rendering and animation. The systems are not designed to operate in real-time and record animations by stop-frame recording techniques similar to those used in traditional animation. Here the emphasis is on the aesthetic, whereas in CADD the emphasis is on accuracy. Packages such as Wavefront and 3-D Studio Max fall into this category.

The final category is modeling for real-time simulation applications. Real-time simulations are meant to be realistic enough by using textures and shading to intuitively visualize the scene content. This allows the user to become immersed in the environment and interact with the surroundings. The models used in real-time simulations are polygonal and optimized for performance. Multigen is an example of a real-time simulation modeling and development package.

Requirements for effective visualization

For a simulation to occur, the geometry of objects must be constructed. To effectively view and interact with objects, the environment must be realistically rendered. Interaction between objects involves some type of behavioral modification of the object. The following sections delve into these concepts more deeply.

Geometry modeling. Geometry can be modeling by a variety of computer programs using points (vertices), lines, and surfaces. Vertices represent a single point on the actual structure. A line connects two points in space. A series of connected lines is usually referred to as a polyline. B-splines or nonuniform rational b-splines (NURBS) could represent complex lines or surfaces. Solid models can be created using the processes of extrusion or lathing. Extrusion is the process of defining a surface or solid by sweeping a curve or polygon along a path in space. Lathing produces a solid by sweeping a curve around an axis.

The geometry of an object is made up of a mesh of polygons that are usually either triangles or quadrilaterals.

Sensory input. A simulation provides sensory information to the user. The better and more complete the sensory feedback is, the more realistic the simulation will seem to the user. Visual information is concentrated upon heavily. Feedback including sound and touch also enhance the simulation.

The visual appearance of the simulation is important to impart a realistic feeling to the simulation. Texture and shading capabilities add realism

to the simulation. The most basic type of shading is flat shading that paints a polygon a single color. Gouraud shading produces smoother shading and probably is a good choice for a simulation. Other shading techniques, such as Phong shading, which add more realism, may affect the performance of the simulation. Other effects, such as transparency, light sources, atmospheric effects, shadows, and reflections, will all add to the realism.

Interaction within the virtual environment. Interaction with objects and their behaviors add realism and may be required for a particular purpose. Collision detection of objects allows the impedance of movement in the environment. Physics-based modeling allows a realistic reaction of an object to a stimulus. The interactions could result in rigid body motions or deformations based upon the properties of the objects. Rigid body motions could result from gravity, friction, bouncing, or sliding. An object could deform after striking another object in addition to a rigid body motion (e.g., a change in trajectory).

Performance issues. The complexity of the model will affect the response time of the simulation. There are methods to help alleviate some of the load on the rendering engine and increase the performance of the simulation.

Modeling of an object is usually performed with polygons. The more complex an object is the more polygons will be required to effectively model the object. Also, the greater the number of polygons used, the greater the computational effort will be required to generate the image. Curved surfaces require more polygons to be effectively modeled. Curved surfaces can be represented by complex polynomials, but in the simulation environment, the representation is converted into an approximation using polygons.

Levels of detail may be used to reduce the load on the system. This technique essentially provides more detail in an object the closer the object is to the viewer. Therefore, an object farther away from the viewer can be modeled with fewer polygons.

Billboarding is another technique employed to minimize polygon count. A realistic tree can require thousands of polygons to model. To decrease the polygon count but present a believable tree, a texture can be applied to a single polygon that rotates to always face the viewer, giving the impression of a solid object. This would allow a forest to be created with realistic looking trees. This technique in the S1000 Toolkit is called a stamp.

Bounding volumes may be used for collision detection, visibility calculations, and view culling. This involves using simpler shaped volumes such as cubes, cylinders, and spheres to represent the volume of a complex object. Therefore, the bounding volumes may be used in collision calculations instead of objects composed of hundreds or thousands of individual polygons.

Models are formed from primitives. The manner in which models are stored in simulation software such as Multigen is in a hierarchical data structure. The data are represented in a tree like structure. For example, a robot would be stored as shown in Figure 5. This structure provides an effective way to manage and manipulate the object. If the body of the robot moves, then the arms and thumbs also move. A movement applied to a higher node affects all lower nodes in the hierarchy. A movement applied to a lower node does not affect nodes higher in the hierarchy.

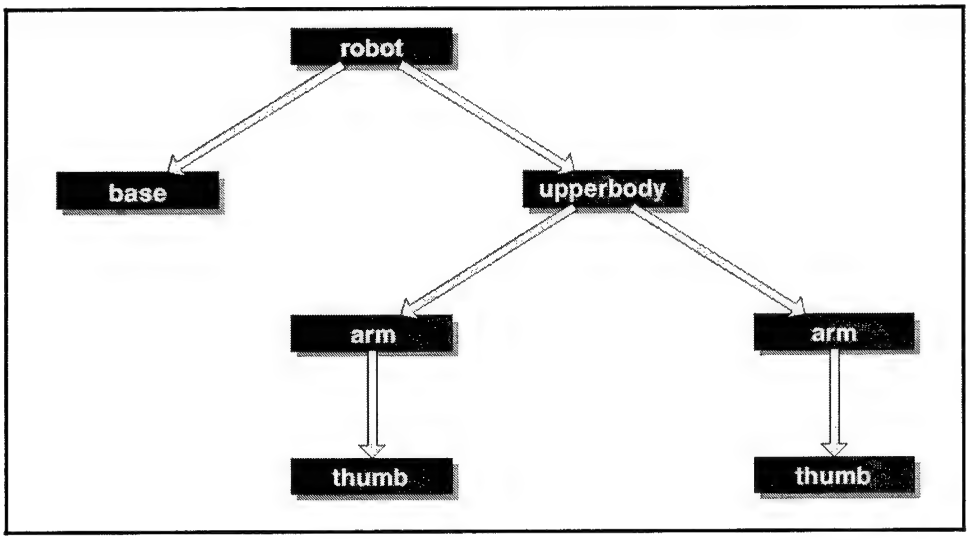

Figure 5. Example of a hierarchical data structure

Instancing is an important technique in modeling. If there are several identical objects in the environment, the object only has to be modeled and stored once. Thereafter, the model may be placed multiple times at various orientations.

Partitioning or zoning is a technique to partition the virtual world into parts that can be displayed independent of each other. For example, different rooms in a house or the inside and outside of a car are different zones. The simulation can switch zones on and off at appropriate times and reduce the amount of information that must be rendered.

4 Graphics Packages, APIs, and Formats for Visualization

Commercial Software

The process described earlier in this report of converting a 2-D paper line drawing to a 3-D object oriented file format would involve the use of many different software tools. The tools that can be used to create the 3-D geometry of a building fall into the following categories:

a. 2-D CADD.

b. 3-D CADD.

c. Modeling, rendering, animation.

d. Raster-to-vector conversion.

e. Format translation.

Other types of software that could be useful for constructing and visualizing buildings are:

a. Real-time simulation software/graphics APIs.

b. Photo modelers.

c. 3-D home design software.

Refer to Appendix B for vendor and pricing information about the software packages discussed in this section.

The Terrain Data Representation Branch of the U.S. Army Topographic Engineering Center (TEC) has put together a large list of visualization software that can be found at *http://www.tec.army.mil/TD/tvd/survey/index.html.*

2-D CADD

Some simple 2-D CADD programs are AutoSketch and TurboSketch. These packages provide drawing and editing functions to produce 2-D drawings. These are basic CADD drawing packages that support drawing with lines, arcs, circles, rectangles, and polylines. Editing functions include operations such as trim, intersect, and extend.

3-D CADD

There are many commercial 3-D CADD packages available. One of the main advantages of using a CADD package for modeling is the accuracy in modeling that can be achieved. CADD packages are built for engineering applications and have tools to aid in the exact placement and sizing of objects. CADD packages have primitives such as lines, polylines, polygons, circles, arcs, rectangles, spheres, cylinders, and cubes that can be used to model. CADD editing tools such as trim, extend, chamfer, and fillet ease the modeling process. Some packages can perform boolean operations on solids or parametric modeling.

Some CADD packages only perform modeling such as Project Architect. Others can perform modeling and rendering such as TurboCAD, TurboCAD SolidModeler, and TurboCAD 3D Modeler. These packages that can render images using textures, shading (Gouraud, Phong, and ray tracing), multiple light sources, and shadows. Other packages such as AutoCAD, MicroStation/J, and ArchiCAD perform modeling, rendering, and animation to produce walkthroughs and fly-bys.

AutoCAD is a popular graphics program for mapping, civil engineering, and surveying applications. AutoCAD supports DXF import and export and other standard formats.

Some programs such as Architectural Desktop, MicroStation Triforma, and ArchiCAD take an object-oriented approach to modeling. Objects such as windows, walls, and doors have some intelligence. That is, when a window is placed in the middle of a wall, the wall knows how to modify its geometry to accept the window. When a wall is drawn in plan view, a 3-D wall is actually being drawn. When walls are drawn in plan, elevation, or 3-D views, the walls know how to draw themselves accordingly.

The increase in power in PC's has lead to a blurring of the distinction between CADD programs and rendering programs. All of the 3-D CADD programs listed above have some capability for 3-D rendering.

Modeling/rendering/animation

There are many programs available to perform modeling, rendering, and animation.

ModelView is only a rendering and animation program. The geometry must be supplied to ModelView. Other programs that perform modeling, rendering, and animation are 3-D Studio MAX, 3-D Studio VIZ, Carrara, Ray Dream Studio 5.5, Animation Master, Softimage 3D, LightWave 3D, Wavefront, and Maya. These programs range in prices from several hundred dollars to several thousand dollars.

These programs have capabilities that include such items as:

a. NURBS.

b. Parametric mesh modeling.

c. Ray tracing (reflections, refractions).

d. Lens effects (flares, glows, highlights, focus).

e. Lighting (specular).

f. Atmospheric effects (clouds, smoke, fog).

g. Particle systems (spray, blizzard, cloud).

h. Particle effects (collision, mutation, trails).

i. Inverse kinematics.

j. Rigid body dynamics (collision, bouncing, sliding).

3-D studio MAX is a popular rendering program that has an impressive list of capabilities. The 3-D Studio VIZ is a product aimed more at architectural applications with capabilities for boolean operations on solids. This provides a relatively rapid method to create complex building geometries from primitive 3-D solids.

Lightwave 3-D is a rendering and scene generation application. The program allows the user to model, render, create surfaces, and animate 3-D graphics from a single interface. The program allows ray tracing and inverse kinematics. The program can import and export 3-D DXF files plus numerous other raster graphics formats.

Maya is a modeling and animation package that includes a node-based scene structure. Maya has NURBS, polygonal modeling capabilities, soft-body dynamics, particle dynamics, and particle rendering.

The purpose of these rendering programs is to produce a photo-realistic rendering of a scene for either a picture or animation. The modeling capabilities of these programs are most important because OneSAF, ModStealth, or MakStealth will perform the rendering for a OneSAF simulation.

Raster-to-vector converters

There are several programs available to convert raster images to vector images. Raster images are a bitmap image that results from scanning in a

paper document (e.g., a floor plan). To be of use in modeling a building, the raster image must be converted to a vector image that is composed of lines.

Some software available to perform this conversion is AutoCAD Overlay 2000, TracTrix, Draftsman PLUS 32, E.G.S. Quick Vector, and Scan2CAD. The price varies greatly on these products and the quality should be closely examined. That is, the more expensive product may not be the best. Conversion of raster to vector images will undoubtedly always need some type of cleanup. That is, the conversion software will probably not reliably convert 100 percent of the information correctly. Even if all information were correctly converted, items such as dimension lines, text annotations, and architectural symbols must be removed.

File format translation

During the modeling process, files may need to be converted from one format to another. There are several programs available to perform translations from one file format to another. These programs are Interchange, Accutrans, PolyTrans and Hijaak Pro. Hijaak Pro converts more than 85 formats including raster and vector formats. Formats including 3-D Studio (`.3ds`), VRML (`.wrl`), IGES (`.igs`), and AutoCAD 3-D DXF (`.dxf`) are converted. The Accutrans, Interchange, and PolyTrans program can convert other useful 3-D formats such as Wavefront (`.obj`), OpenFlight (`.flt`), 3-D Studio Max (`.max`), Lightwave (`.lw`), and trueSpace (`.cob`, `.scn`).

Real-time simulation/graphics APIs

Software for real-time simulation provides the capability for the user to interact inside of a virtual world. Software in this category shares many commonalties with rendering and animation software. Not all of the simulation software has 3-D modeling capabilities. The models can be given certain behaviors and can be viewed and manipulated in 3-D in real-time. The models used in this type of software are composed of triangles and quadrilaterals. Some of the software while also possessing modeling and rendering capabilities can be used in programming languages to build applications.

MakStealth is a viewer that delivers a realistic 3-D view of the terrain, vehicles, and other entities in a virtual world. It is compatible with the Multigen Openflight file format. The viewer supports the High Level Architecture (HLA) and Distributed Interactive Simulation (DIS) protocols. The stealth can display smoking and flaming effects and muzzle flashes. The stealth presents the terrain realistically and lets the user control fog, time of day, and other effects. The simulation may be viewed from several different viewpoints.

Multigen Creator is a toolkit for creating highly optimized, high fidelity real-time 3-D content for use in simulation applications. Multigen has tools to facilitate the modeling of 3-D objects such as tanks, aircraft, and buildings. A digitizing tablet can be used to input information from maps or drawings. Multigen can generate terrain polygons from gridded data such as USGS DEMs or National Imaging and Mapping Agency (NIMA) digital terrain elevation data (DTED). Multigen also allows the use of NIMA Digital Feature Analysis Data (DFAD). Multigen supports levels of detail with morphing, textures, instancing, 3-D sound, view culling, lighting, collision detection, and extrusion. Various file formats can be imported and exported including 3-D Studio (`.max`), 3-D Studio MAX (`.max`), IGES (`.igs`), DXF (`.dxf`), and Wavefront (`.obj`). The OpenFlight file format is in the public domain. Multigen can produce correlated CTDB data from OpenFlight databases.

WorldToolKit is a toolkit for building 3-D visual simulation and virtual reality applications. The toolkit consists of object-oriented libraries that supports geometry objects, sensors, lights, collision detection, atmospherics, texture mapping, 3-D sound, materials, translucency, and more. The toolkit supports real-time 3-D rendering.

3DLinX is a software development environment designed to make realistic 3-D applications easy to create. 3DLinX is a real-time graphics rendering engine and development tool based on Microsoft's Component Object Model (COM) architecture and ActiveX technology. 3DLinX is an ActiveX control that can be used in languages such as Visual Basic, Visual C++, or Delphi. The control automatically handles graphics rendering optimizations, scene graph management, collision detection, view culling, and multithreading. Also, 3DLinX can apply textures, transparencies, reflections, shadows, lights, and 3-D sounds. The control directly imports 3-D Studio and OpenFlight models with textures.

SuperScape VRT is a software development kit (SDK) for development of virtual worlds. It has an extensive collection of prebuilt objects with behaviors, textures, and sounds. The SDK supports texture mapping, texture projection across multiple planes, real-time environment mapping, shadows, mirrors, lens flares, shiny surfaces with specular highlights, transparency, real-time lighting, collisions, gravity, friction, automatic level of details, and more. An ActiveX control is available for use with languages such as Visual Basic, Visual C++, or Delphi. A plug-in for web browsers called Viscape Universal is also available to view models over the web.

The Java 3-D API is an application programming interface used for writing stand-alone 3-D graphics applications or Web-based 3-D applets. Java 3-D has high level constructs for creating and manipulating 3-D geometry and tools for constructing the structures used in rendering that geometry. With Java 3-D very large virtual worlds can be designed and rendered. Java 3-D is designed as a high-level platform-independent 3-D graphics programming API and is amenable to very high-performance implementations across a range of platforms. Java 3-D's scene-graph based model makes it ideal for

Virtual Reality systems and other applications that wish to represent and navigate complex 3-D worlds. More information on Java 3-D can be found at *http://java.sun.com/products/java-media/3-D/index.html.*

Bamboo is a toolkit for virtual environments being developed by the Naval Postgraduate School. Bamboo is based on OpenGL.

A listing and overview of about 600 3-D engines for real-time graphics and virtual environments can be found at *http://cg.cs.tu-berlin.de/~ki/engines.html.*

Photo modelers

A specialized type of software exists that allows the user to produce a realistic 3-D model from photographs. Textures from the photographs can be used to add realism to the 3-D model. The programs can be used to produce walkthroughs, fly-bys, take measurements of hard-to-reach areas, and visualize proposed architectural changes. Programs with this capability include Canoma, PhotoModeler, and 3-D Builder Pro.

Basically, photographs are used as a template. Primitives such as cubes and prisms are overlaid on the photograph. The corners of the primitives are pinned to locations on the photograph overlaying objects such as a building. From the position and orientation of the primitive, the correct perspective is computed for the photograph. Additional photographs can be used to supply geometry and texture information from various locations and angles. The programs automatically take the textures from the photographs and apply them to the 3-D models.

Home design software

There is a specialized class of software for the design for homes. This software helps to quickly design and layout a home floor plan. The home is modeled in 3-D and can be rendered with animated walkthroughs. Some capabilities supported by these packages include scanning of floor plans or drawings and conversion to a digital format, automatic roof generation, multilevel walkthroughs, smart objects (e.g., doors, windows) to aid in object placement, and furniture libraries. Some packages support export of the model to the DXF format, while other packages can export to the VRML format. Many packages are available on the market and include 3D Home Design Suite Deluxe, Punch! SUPER Home Suite, FloorPlan 3D Design Suite, and Planix Complete Home Suite.

Government Software

The following graphics packages are available from the Government that can aid in the process of producing a building in the required format for the OneSAF program.

S1000 Toolkit overview

Lockheed Martin Information Systems under contract with TEC developed the S1000 Toolkit. This is the main software package that TEC uses to create terrain databases. The objects in the 3-D database are built up using polygons and assigned color and physical attributes. The S1000 Toolkit uses triangles and quadrilaterals to represent physical objects. Polygons are given attributes such as color (on one or both sides) or textures using texture maps. Texture maps enable a more detailed appearance without requiring detailed modeling with polygons. The textures can represent such items as weapons effects, building facades, and paved highways. Objects such as roads, railways, runways, and streams are represented by linear feature data. Treelines and canopies can represent large quantities of foliage. A canopy consists of vertical polygons textured with tree shapes that form an enclosure. Stamps are 2-D polygons that rotate to face the viewer at all times. This produces a 3-D effect (TEC 1996).

Three-dimensional objects such as buildings, single trees, towers, and vehicles are created with the S1000 modeling tool and placed on the terrain as either static or dynamic objects (TEC 1996). Each model is given an associated bounding volume that is used for ballistics and collision detection. Models can also have multiple levels of detail representing various levels of complexity of the model when viewed at certain distances.

S1000 combines terrain data from a variety of sources. The NIMA is the most used source of terrain data for Department of Defense (DoD) simulations. NIMA's DTED, interim terrain data (ITD), and digital feature analysis data (DFAD) are used to develop the terrain skin and terrain feature models. The DTED is a digital elevation model characterized as a grid of elevation values for a specific geographic area. The ITD was developed by NIMA as an interim product for digital terrain analysis until the Tactical Terrain Data could be fielded in its final format. The ITD is the primary source data set for cultural/terrain feature models. The DFAD is another source of terrain feature and cultural data. DFAD is used where ITD data are lacking or nonexistent. DFAD is the primary source for urban areas (Welch in preparation).

Geometry can be imported from commercial packages using the DXF data format. The S1000 source data are compiled to form the run-time databases for the SAF PVD, the ModStealth 3D display, and the dynamic terrain object database (Figure 6).

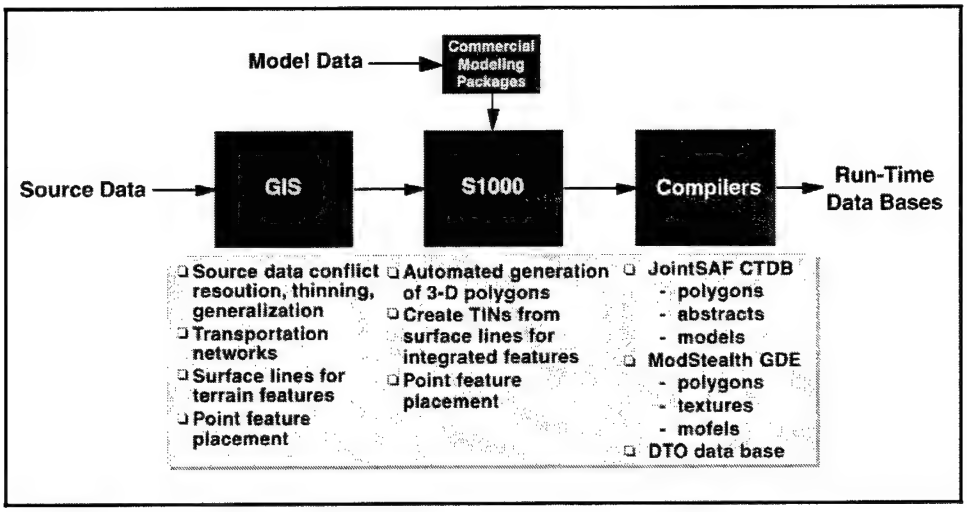

Figure 6. Overview of database generation (from LMIS 1997)

At the present time, the S1000 Toolkit can model bridges as MES structures. Building cannot be modeled as MES structures within S1000. Another tool is necessary to construct the MES building.

Building geometry can be modeled in S1000 using the Modeling Tool and placed on the terrain by using the Assembly Tool. These buildings would provide the geometry needed for the visual database (ModStealth 3-D display).

MES Toolkit overview

The information contained in this section is taken from the DWN MES Creator Developer's Guide (SAIC 1998). The MES Toolkit was produced as part of the Dismounted Infantry (DI) SAF program. SAIC ASSET GROUP ADST II developed the MES Toolkit during the period January 1996 to March 1999.

SAIC produced a plug-in for AutoCAD that produces information necessary to construct an MES building. The plug-in reads in a data file in OpenFlight format and produces a OneSAF reader file (.rdr file) that can be used to insert the MES building into a CTDB terrain database using the terrain recompile program. Multigen or any program that can produce OpenFlight files can produce the OpenFlight files. The plug-in supports version 14.2 OpenFlight files.

Modeling in AutoCAD

Each layer in AutoCAD corresponds to a separate enclosure or aperture. Enclosure layers are always named starting with *ENC* and cannot contain underscores. The exterior enclosure layer must be named `Exterior`.

Names are case insensitive. Valid names are `ENC10`, `Exterior`, `Enc-TopBack`, or `Enclosure`.

Aperture layers start with *Ad* for a door or *Aw* for a window. The next two components of the aperture name are separated by an underscore and designate which two enclosures the aperture connects. An optional tag may be used at the end of the aperture name if more than one aperture exists for an enclosure. The naming convention for a window would looks like `AW_<encl1>_<encl2>_<optionalTag>` and a door looks like `AD_<encl1>_ <encl2>_<optionalTag>`. Valid names are `AW_exterior_enclosure1`, `AD_ENC1_ENC3_A`, and `AD_ENC1_ENC3_B` assuming `enclosure1`, `ENC1`, and `ENC3` are names of enclosures.

The AutoCAD plug-in exports only 3DFACE objects. Other drawing primitives may be used as templates to aid in constructing the MES. Any OpenFlight files that are used cannot contain external references, or any type of transform or level-of-detail bead in the database hierarchy. All geometry loaded from the OpenFlight file is represented as 3DFACE objects. Typically, the OpenFlight file will not be in the format needed to construct a MES directly and can only be used as a template for recreating the building geometry in the required format.

Apertures have two thicknesses associated with them and a direction or front-face. In order for apertures to be drawn properly on the OneSAF PVD display and for them to be processed correctly, there are four things that must be correctly modeled. This assumes that one enclosure butts against another and that the aperture is drawn right on the edge where the two enclosures meet:

a. The aperture should be drawn on the edge where the two enclosures meet (Figure 7).

b. The aperture should face INWARD toward the enclosure that contains the frame portion of the window or door(way).

c. The aperture should be named such that the first enclosure in the aperture layer name is the enclosure that the aperture polygon faces.

d. The `wall_thickness` parameter in the generated `.rdr` file should be modified to represent the thickness of the frame-area for the FIRST enclosure in the "`connected_to`" list for each aperture.

The enclosures referred to in steps *b*, *c*, and *d* are all the same enclosure for a given aperture connection.

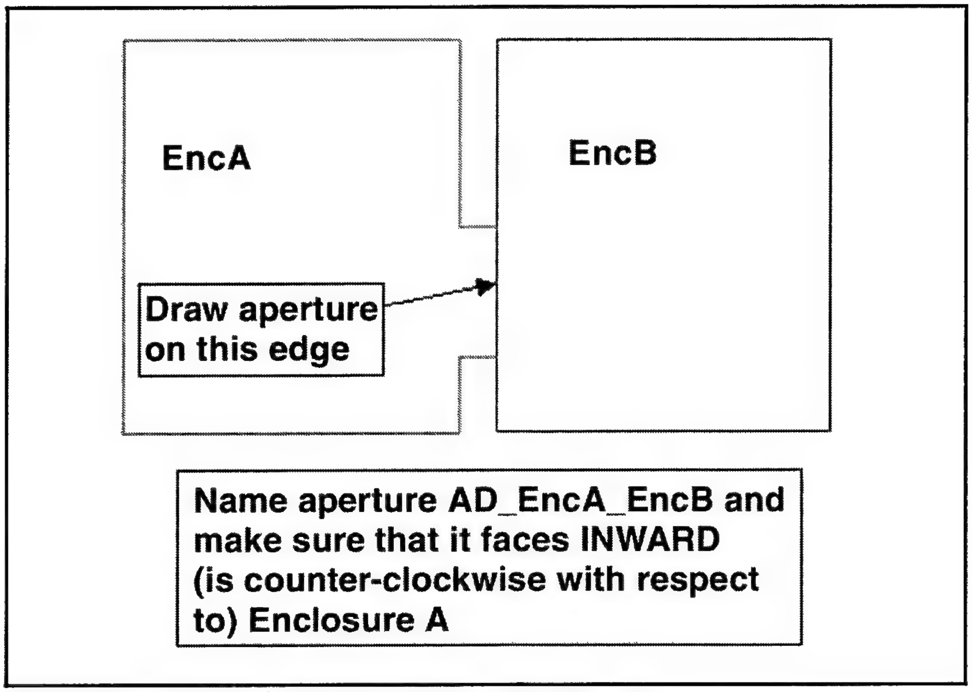

Figure 7. Construction of an aperture

Making the reader file and modifying the CTDB

Once the model is completed, the plug-in will write out a reader file for use in the terrain recompile program. Some manual editing of the reader file is required before the recompile program can be used. The reader file has two sections, a template section and a volume section. The volume section tells the compiler where to instantiate the building and at what orientation. The same template can be instantiated multiple times by including multiple volume sections in the reader file. The plug-in creates the template section, but the volume section must be created by hand. The template and volume sections were discussed earlier in this report.

Other changes to the reader file include specifying the:

a. Material type of the aperture - doors are normally material 119 and windows are material 40.

b. Wall thickness - defaults to 0.001 and 0.000. The value on the side of the aperture facing into the frame area should be 0.20 to correspond to average wall thickness.

c. Model name - defaults to Building A.

After the manual changes are complete, the reader file can be recompiled into an existing CTDB. The CTDB recompile program is used to accomplish this.

Data Representation Formats

There are many different file formats available to represent the geometry and physical appearance (i.e. texture and lighting) of an object. Some formats are capable of only representing the geometry while other formats can be very complex including information on textures, lighting, sound, levels of detail, and more. It will be important to select a suitable format in which to represent the building data as the modeling process progresses from stage to stage. The formats used may ultimately be dependent upon the graphics packages used in the modeling process. Whatever the formats used, the final information must be able to be converted to a format that can be compiled into the CTDB and 3-D visual databases.

The following are examples of some file formats available that can be used to model buildings.

3-D DXF (`.dxf`)

The DXF format is the interchange file format for Autodesk's AutoCAD package. The format has become a widely accepted format for the transfer of 3-D geometries between different packages, not necessarily involving AutoCAD. For any given renderer, a DXF import facility usually exists. DXF includes support for 3-D objects, curves, text, and associative dimensioning (Gallop et al. 1997).

The DXF format is an ASCII format that is used to store geometry information. Information regarding textures, materials, and lighting is not stored in this format. Therefore, geometry can be transferred by this format, but texture information would need to be added or redone in another software package.

IGES (`.igs`)

IGES, the Initial Graphics Exchange Specification, is a widely used CAD data exchange specification. However, IGES is a very complex specification and fully utilizing IGES data files requires expert knowledge of the format. An overview of the IGES validation service at National Institute of Standards and Technology (NIST) is available. Many CAD systems import and export IGES (Gallop et al. 1997).

3DML

3DML is a based on the Extended Markup Language (XML). 3DML runs on web browsers using the Flatland browser plug-in. A 3-D world called a Spot can be made fairly easily using predefined blocksets. A simple ASCII map that lays out blocks representing walls, floors, ceilings, stairs,

furniture, and many other items provides the layout of the world. The world layout must be a regular grid of blocks. 3DML supports directional sound, hotspots, and avatars. Information on 3DML and the browser plug-in can be found at *http://www.flatland.com.*

VRML (`.wrl`)

VRML (Virtual Reality Modeling Language) is a platform-independent file format for sharing 3-D worlds on the Web. VRML worlds can be interactive and animated, and can include embedded hyperlinks to other Web documents. VRML 97 is International Standard ISO/IEC 14772-1:1997 (VRML 1997). Some features of VRML 97 include:

a. Sound objects with controllable attenuation.

b. An efficient system to describe irregular ground terrains.

c. Extrusion objects for advanced but compact modeling.

d. A fog system allowing underwater and cloudy environments to be represented.

e. The ability to use MPEG video as a texture map.

f. Levels of detail.

g. Collision detection of objects.

h. Touch sensors that allow reactions to a users deliberate actions.

i. Proximity sensors that allow reactions to a user's not-so-deliberate actions.

VRML is a simple ASCII text description of geometries that can be shared across platforms (UNIX, mac, and windows). VRML is quite difficult to author.

Wavefront (`.obj`)

This `.obj` file format for the Wavefront rendering program from Alias|Wavefront is very popular. The file is an ASCII file that describes the geometry, textures, and lighting necessary to render a scene.

OpenFlight (`.flt`)

Another popular format for simulations is the OpenFlight format from Multigen. This format is designed to support both simple and relatively sophisticated real-time software applications. The format supports variable levels of detail, degrees of freedom, sound, instancing, replication, animation sequences, bounding boxes for real-time culling, shadows, advanced

scene lighting features, lights and light strings, transparency, texture mapping, material properties, and other features.

The OpenFlight database hierarchy allows the visual database to be organized in logical groupings and is designed to facilitate real-time functions such as level-of-detail switching and instancing. The database is organized in a tree structure.

SEDRIS

SEDRIS, the Synthetic Environment Data Representation and Interchange Specification, is a Defense Modeling and Simulation Office (DMSO) sponsored effort to develop and promulgate a standard data model and API supporting the distribution and reuse of environmental data within the M&S community.

A common representation of the physical environment is necessary for heterogeneous simulations to interoperate. The SEDRIS data standard was developed to allow the interchange of environmental data between heterogeneous simulations. This standard promotes reuse of data and alleviates problems associated with data completeness and consistency (SEDRIS 1999).

The SEDRIS project was conceived and implemented to capture and provide a complete data model of the physical environment, access methods to that data model, and an associated interchange format.

This neutral file storage of data allows the conversion from one format to another. An application that uses this format would only need to be able to read and write this format to convert from one application to another. This alleviates writing specialized converters to transfer information between desired applications.

Sources for file formats

A good source for file formats can be found at *http://www.wotsit.com* and *http://www.cica.indiana.edu/graphics/3D.objects.html.*

5 Technical Challenges

There are many technical challenges that must be overcome if a visual walkthrough of a building is to be developed within 36 hr of receipt of the structural information. These challenges can be categorized into challenges associated with engineering, data, and software. The following sections describe these challenges in more detail.

Availability of As-builts

Floor plans, architectural rendering, elevations, photos, or other information detailing the form and layout of a building will be required to enable a 3-D model to be constructed. The information contained on original blueprints may not be reliable. As-built drawings are a better source of information but may be nonexistent. There may even be a scarcity of any kind of information concerning the layout of the building. Therefore, the amount, quality, and accuracy of information will affect the final building model and ultimately the usefulness of the model. If the available information concerning the interior layout of walls is not reliable, the model will be of little use for an actual terrorist situation.

Data Migration

Migration of data from the starting point of a blueprint (paper drawing) to the generation of the databases necessary to view a building in a simulation in OneSAF or ModStealth will require the manipulation of data at various intermediate points. As data are transferred from point to point in the modeling process, the data integrity and visual representation must remain faithful to the information provided by the user.

Data Formats

The format of the data that will be used at each stage in the modeling process must be capable of recording all of the pertinent structural information at that particular stage of modeling. The format must be capable of carrying the information to the next stage for further processing. If COTS packages are used, several formats may be involved as the modeling process proceeds because not all packages support the same formats. The DXF format is a format that is supported by most COTS and GOTS packages. However, the DXF format is only capable of defining geometry. To define textures, which are desirable, another file format would be required. The DXF format is also not a standard format. The format varies with each release. This could result in data integrity problems when transferring data from one software package to the next.

Some formats that are capable of describing both the geometry and textures are the VRML, OpenFlight, OBJ, and 3DS formats. These are popular formats available in COTS and GOTS software packages. One of these formats may be suitable as a neutral file format for storing and converting data.

The S1000 data formats are used by the terrain compilers to produce the simulation databases. It is desirable to use the S1000 data formats if possible to represent the MES buildings. This may have to be done external to the S1000 program.

The SEDRIS data format shows promise as a neutral file format capable of storing all needed information for use in other packages. Presently, the SEDRIS format is not as widely supported as the other formats. The SEDRIS format is being developed as a means to transfer all virtual environment data from one simulation to another.

Recognition of Building Features

Data can enter the modeling process in the form of a paper drawing. As discussed in Chapter 3, there are several alternatives of converting this analog format into a digital one. The process may involve manual or automatic conversion into a 2-D digital format.

If an automatic conversion process is used, it must be able to identify the features of interest. Identification of these items of interest from a detailed engineering drawing will be a significant challenge in the process. Initially, the walls, windows, and doorways are the most important features to recognize. These features form enclosures and apertures that are needed to define the simulation databases. As the process matures, other features could be added such as lighting, plumbing, and HVAC systems. Mobility limiting features such as furniture will be one of the last items added.

Drawing features such as dimension lines, text, and architectural symbols must be discarded to produce a "clean" view of the floor plan. The cleanup process will inevitably need human intervention to identify objects that are not of interest.

If a 2-D CADD drawing is used as the initial data, the task of identifying structural features will be easier. The format of a CADD file is expected to be the DXF format. This format contains metadata for describing geometry, text, and dimension lines. Also objects within the file may have an identifying name that will help determine the structural feature. Most COTS packages can import and export the DXF format.

Software Operational Limitations

Presently, the S1000 Toolkit does not directly support MES buildings. That is, the data structures necessary to represent an MES building cannot be modeled within S1000. The 3-D visual geometry of the building can be modeled within S1000. The MES Toolkit is the only software available at this time that can produce the data structures necessary to define an MES building. MES buildings are important in OneSAF because this type of building allows movement and operations within the building interior.

The S1000 Toolkit can produce the 3-D geometry with textures needed for the 3-D visual database to operate ModStealth. The MES Toolkit only aids in the production of the CTDB database required for OneSAF. Therefore, both tools perform a portion of the total modeling effort required to visualize a building in OneSAF and ModStealth.

Using both the S1000 and MES Toolkits, an MES building can be modeled and placed on a terrain. The problem is the effort is labor intensive, time consuming, and subject to error. The process of producing data files that the terrain compilers can use must be streamlined. This may require use of additional GOTS, COTS, or specially developed software to speed up and simplify the modeling process.

The OTB OneSAF software is released for testing purposes. Therefore, the program may contain unstable portions of code that do not perform the intended function. The rapid addition of an MES building to the terrain will not prove useful if the MES buildings or multistate objects do not perform as intended. A previous version may be used if it contains code that functions correctly. If a previous version has to be used, this could hinder integration into the final product. The correct operation of MES buildings, multistate objects, and dismounted infantry inside of the OneSAF program is outside the scope of work to rapidly model buildings.

Time Constraints

The use of present GOTS packages to produce the data needed for viewing buildings in OneSAF and ModStealth is too time consuming. The goal is to produce a building in the OneSAF environment for mission rehearsal and initiate the mission within 72 hr of receiving the structural data. Therefore, the building model must be completed within approximately 36 hr to allow time for mission rehearsal.

The complexity of the building will affect the amount of modeling time required. A 1-story office building will require much less time than a 10-story apartment complex. Also, the level of detail required in the model will affect the modeling time. The modeling time will increase as more details such as doors, doorknobs, furniture, sewer lines, and HVAC systems are required.

Response Time of Simulation

The response time of the simulation will be dependent upon the level of detail or fidelity of the model. A finely detailed model may convey a very realistic feel for the simulation but degrade the system performance to an unacceptable level. The detail of the model must be keyed to the purpose of the simulation. Enough detail should be modeled to satisfy the simulation requirements, but not too much to adversely affect the response time. The level of detail will vary with each application.

Effective Modeling

Constructing an MES building is difficult and labor intensive. Birkel and Lukes (1998) state that MES buildings are too complex to reliably generate by hand. A process must be developed to streamline the construction of MES buildings for use in OneSAF and ModStealth. The present method of modeling MES buildings is not synthesized into one software package. A viable solution may involve several software packages including GOTS, COTS, and specially developed data converters or modeling packages. The process must be able to quickly assimilate the information from blueprints, add visual detail, and output the building information in a format that the terrain compilers can use. This process may involve automatic conversion of scanned drawings, intelligent placement of building objects such as doors and windows, and methods to accelerate the geometry modeling. The process needs to make the most efficient use of GOTS and COTS software available to avoid unnecessary software development. The modeling process must be able to define a building of sufficient complexity to be able to represent most real world buildings.

The building model must be able to also accept structural attributes necessary to perform structural analyses involving blast effects. This capability should be planned for as the modeling process is developed.

Correlated Databases

The construction of the simulation databases start from the same set of initial data (e.g., a blueprint) but proceed along different paths going through different software packages and data conversions. This results in two databases with the potential for data inconsistencies. The databases are not correlated. It would be desirable for the modeling process to proceed from the same initial data set and end with a data set that contained both the MES building and 3-D visual information.

Integration with Multistate Objects

The method of producing MES buildings should be compatible with multistate objects to allow damage states of buildings to be computed and displayed.

6 Recommendations for Future Work

The following recommendations are made regarding continued work on the development of a capability to rapidly model buildings for use within the OneSAF software:

a. Verify the process of producing an MES building using the MES and S1000 Toolkit. This will also allow the verification of the format needed for the reader file.

b. Explore the usage of the MES toolkit and S1000 Toolkit to identify weaknesses and places for improvements.

c. Further investigate COTS packages to identify the packages that are best able to help in the modeling process.

d. Determine if the GOTS packages can perform the required modeling in the most efficient manner. Determine how the modeling process using GOTS packages can be improved and if additional software must be developed.

e. Investigate the possible modification of S1000 to include native support for MES buildings. This would not be an immediate task since it is viewed as being labor intensive. More collaboration with TEC and Lockheed Martin Information Systems about the S1000 Toolkit should enable a determination of this possibility and the identification of other areas that need improvement.

f. Investigate any additional work being performed by SAIC on the MES Toolkit.

g. Review the file formats of the GOTS and COTS packages to identify the best formats to use in the modeling process. The SEDRIS format should be evaluated as a neutral file format to store the building information.

h. Investigate the definition of multistate objects to determine how they can be integrated with the MES buildings. Additional data required and modeling procedures to construct multistate objects should be investigated.

i. If at all possible, produce correlated databases to reduce the duplication of effort and possibility of data inconsistencies.

j. Begin the modeling process examining basic geometry (i.e., orthogonal walls) and then proceed to more complex geometries.

k. Examine methods to accelerate the modeling process such as predefined libraries of building parts or furniture libraries.

References

Birkel, Paul, and Lukes, George. (1998). *Synthetic environments*, Prepared for the Defense Advanced Research Projects Agency (DARPA) Synthetic Environments Program, Document #98-5-3100, Washington, DC.

Costello, P., and Bee, S. (1997). " 3D model databases: The availability of 3D models on the World Wide Web," Advanced VR Research Centre, Loughborough University of Technology, UK.

Gallop, Julian, Osland, Chris, Haswell, Janet, and May, Roger. (1997). "AGOCG rendering evaluation," Advisory Group on Computer Graphics, UK.

Lockheed Martin Advanced Distributed Simulation (LMADS). (1999a). *Functional description document (FDD)*, Lockheed Martin Advanced Distributed Simulation, a business unit of Lockheed Martin Federal Systems, ADS Document Number: ADST-II-CDRL-MODSAF5.0-9800327.

______. (1999b). *User's Manual for ModSAF v5.0, Volume 1*, Lockheed Martin Advanced Distributed Simulation, a business unit of Lockheed Martin Federal Systems, ADS Document Number: ADST-II-CDRL-MODSAF5.0-9800324, CDRL No. AB04.

Lockheed Martin Information Systems (LMIS) (1997). *Southwest Asia terrain data base design document synthetic theater of war*. Advanced Simulation Laboratory, Bellevue, WA.

______. (1998). *Final report: Dynamic terrain and objects within a virtual world.* CDRL Sequence Number A004 and A003, Advanced Simulation Center, Bellevue, WA.

______. (1999). *Software architecture design and overview document (SADOD) for ModSAF.* Produced by Lockheed Martin Information Systems for STRICOM, Contract Number N61339-96-D-0002/D.O. 75, ADS Document Number: ADST-II-MODSAF5.0-9800323, CDRL No. AB0.

Lukes, George. (1998). "Synthetic theater of war," slide presentation, Defense Advanced Research Projects Agency, Washington, DC.

One Semi-Automated Forces (OneSAF) Operational Requirements Document (ORD), Version 1.1 (ORD), 16 August 1999, *www.onesaf.org*.

Reece, Douglas, and Dumanoir, Paul. (1998). "Conventions for representing humans in a DIS exercise: The DWN experience," *Proceedings of the 1998 simulation interoperability workshop*, IEEE.

Reece, Douglas, Ourston, Dirk, Kraus, Matthew, and Carbia, Ivan. (1998). "Updating ModSAF for individual combatants: The DI SAF Program," *Proceedings of the 7th conference on computer generated forces and behavior representation*, Institute for Simulation and Training, University of Central Florida.

Science Applications International Corporation (SAIC). (1998). "DWN MES creator developer's guide," MES Toolkit, available from TEC.

SEDRIS. (1999). SEDRIS web site, *www.sedris.org/overview.htm*.

Stanzione, T., Evans, A., Buettner, C., and Chamberlain, F. (1996a). "Multiple elevation structures in the improved computer generated forces terrain database," *Proc. sixth conference on computer generated forces and behavioral representation*.

______. (1996b). " Multiple elevation structures in the improved computer generated forces terrain database," *Proc. fifteenth workshop on standards for the interoperability of distributed simulation*s, Paper 96-15-119.

Topographic Engineering Center. (1996). *Terrain database design specification, STOW Camp Pendleton terrain database, Version 1.0*, U.S. Army Topographic Engineering Center Digital Products Center, Data Support Division, Alexandria, VA, Sponsored by Defense Advanced Research Projects Agency.

VRML Consortium. (1997). The virtual reality modeling language, International Standard ISO/IEC 14772-1:1997.

Welch, Michael R. "Synthetic environmental databases," Orion Development Group, Inc., in preparation.

Appendix A
3-D Graphics Terminology

Most of the definitions in this appendix were taken from the following sources on the internet:

- *www.stb-international.com/TIS/htmlglossary/HTMLGlossary-Start.htm*
- *www.fastgraphics.com/basics/abc3d.html*
- *www.trid.com/html/3d_graph_term.htm*
- *www.zdnet.com/pcmag/features/software/1519/3d-s4.htm*

3-D Engine - A group of software routines that render polygonal models overlaid with texture maps to create the illusion of a world in three dimensions.

3-D Graphics - Displayed representation of a scene or an object that appears to have three axes of reference: height, width, and depth (x, y, and z).

3-D Pipeline - The process of 3-D graphics can be divided into three stages: tessellation, geometry, and rendering. In the tessellation stage, a described model of an object is created, and the object is then converted to a set of polygons. The geometry stage includes transformation, lighting, and setup. The rendering stage, which is critical for 3-D image quality, creates a two-dimensional (2-D) display from the polygons created in the geometry stage.

Alpha Buffer - An extra color channel to hold transparency information; pixels become quad values (RGBA). In a 32-bit frame buffer, there are 24 bits of color, 8 each for red, green, and blue, along with an 8-bit alpha channel.

Alpha-blending - Alpha-blending is the ability to give a pixel a value that will render it opaque, transparent, or translucent. That is to say solid, invisible, or partially invisible, respectively. This process is commonly used to depict special effects like explosions, water, and glass.

Alpha-transparency - A hardware technique that supports true transparency and translucency by blending the values of a source pixel and a destination pixel allowing for more sophisticated effects. For example, in a car-driving scene, the blue tint on a windscreen would modulate the colors of the sky and landscape beyond.

Animation path - An editable line that objects follow during the course of an animation.

Anti-aliasing - Also known as oversampling, this process reduces, or smoothes, the stair-stepping visual effect often seen in bitmapped graphics.

Atmospheric Effect - Effects, such as fog and depth, that improve the rendering of real-world environments.

Avatar - The abstract representation of the user in a virtual environment. The physical dimensions of the avatar are used for collision detection and terrain following.

BitBLTs - The BitBLT is the single most important acceleration function for windowed graphical user interface (GUI) environments. A BitBLT is simply the movement of a block of data from one place to another, taking into account the special requirements and arrangements of the graphics memory.

Bitmap - A bitmap is a pixel by pixel image.

Blending - Blending is the combining of two or more objects by adding them on a pixel-by-pixel basis.

Bounding Volumes - Bounding volumes such as cubes, cylinders, or spheres, are used to aid in collision detection and view culling. The volumes are placed around more complex objects. The rendering engine can use the simpler bounding volume for collision detection, visibility calculations, or view culling than the more complex objects to speed up rendering.

Bump-mapping - Bump-mapping is a visual trick that attempts to demonstrate roughness or smoothness of a texture usually through use of two individual textures, one actual texture to be mapped to the object and another that is used to determine how rough or smooth the texture will appear.

Camera - In 3-D graphics, the camera is the viewpoint through which a scene is viewed. Fly-throughs of scenes are conceptually a moving camera.

Chroma Keying - Chroma Keying or texture transparency is the ability to recognize a key color within a texture map and make it transparent

during the texture mapping process. Since not all objects are easily modeled with polygons, chroma keying is used to include complex objects in a scene as texture maps.

Collision Detection - The ability of a 3-D object to physically react to another object in space and time.

Culling (Clipping) - Culling (or Clipping) is support for overlapped objects on the screen. Removing from the processing pipeline points and surfaces that are outside the field of view (known as the "viewing frustum"). Culling of objects that are facing away from the viewer is called backface culling.

Depth Cueing - Used in conjunction with fogging, depth cueing is the adjustment of the hue and color of objects in relation to their distance from the viewpoint. Depth cueing is the reducing of an object's color and intensity as a function of its distance from the observer. For instance, a bright, shiny red ball may look duller and darker the farther away it is from the observer.

Dithering - When an image's color depth is lowered from a higher count to a lower count, it is said to be dithered down. When this occurs, any lost color data may be seen by the naked eye as dotted patterns or artifacts.

Double Buffering - By having twice the amount of memory on a graphic board than is actually necessary for a specific resolution and color depth, game performance can be increased by drawing the complete next picture to the buffer that is currently off-screen and then flipping the page. You will always instantly see the next frame and never the process of drawing it. This makes animations smooth and flicker-free.

Dynamic Attachment - Attaching a skin, or object, to another object is known as dynamic attachment. Wrapping a soft skin around the skeleton of the 3-D object avoids the problem of showing joints where the edges of polygons meet.

Environment Mapping - A texture's ability to accurately map a reflection of its surrounding environment on itself is known as environment mapping. This is useful for giving textures a highly reflective appearance, in simulating chrome for example.

Extruding - Drawing out the contour of a 2-D path or shape into 3-D space by extending it in a plane perpendicular to that of the 2-D path.

Flat Shading - The flat shading method is also called constant shading. For rendering, it assigns a uniform color throughout an entire polygon. This shading results in the lowest quality, an object surface with a

faceted appearance and a visible underlying geometry that looks 'blocky.'

Fog - Fog is the blending of an object with a fixed color as its pixels become farther away from the viewpoint. This creates a simulation of distance and atmosphere. The farther the object or terrain is, the higher likelihood it will be enveloped in fog.

Gouraud Shading - Also known as smooth shading. This shading applies a lighting calculation to each vertex of a polygon face, and linearly interpolates the results across the face to achieve a smooth lighting effect with gradual color transitions.

Hidden Surface Removal - Hidden surface removal or visible surface determination entails displaying only those surfaces that are visible to a viewer because objects are a collection of surfaces or solids.

Interpolation - Interpolation is a mathematical way of regenerating missing or needed information. For example, an image needs to be scaled up by a factor of 2, from 100 pixels to 200 pixels. The missing pixels are generated by interpolating between the 2 pixels that are on either side of the pixel that needs to be generated. After all of the 'missing' pixels have been interpolated, 200 pixels exist where only 100 existed before, and the image is twice as big as it used to be.

Inverse Kinematics - In an object hierarchy where there are parent and child objects, grabbing one child object at the end of a chain and automatically calculating the proper movements back to the first object, all according to a series of preprogrammed constraints is known as inverse kinematics. An example would be an articulated hand, where moving the tip of a finger causes all the other parts to move together in a properly jointed way.

Jaggies - Also known as aliasing or the jagged visual appearance of lines and shapes in raster pictures that results from producing graphics on a grid format. Increasing the sample rate in scan conversion can reduce this effect.

Keyframes - User-specified frames that are recorded on an animation time line. The animation application automatically fills in the intermediate action between these frames to create fluid animation.

Lathing - Creating a 3-D surface by rotating a 2-D spline around an axis.

Level of Detail - The amount of detail or complexity that is displayed at any particular time for any particular object. Whatever the level of detail may be for an object varies as a function of the distance of the object from the viewer.

Lighting - There are many techniques for creating realistic graphical effects to simulate a real-life 3-D object on a 2-D display. One technique is lighting. Lighting creates a real-world environment by means of rendering the different grades of darkness and brightness of an object's appearance to make the object look solid.

Mapping - Placing an image on or around an object so that the image is like the object's skin.

Modeling - The process of creating free-form 3-D objects.

Mesh Model - A graphical model with a mesh surface constructed from polygons. The polygons in a mesh are described by the graphics system as solid faces, rather than as hollow polygons, as is the case with wireframe models.

MIP Mapping - A method of increasing the quality of a texture map by applying different-resolution texture maps for different objects in the same image, depending on their size and depth. If a texture-mapped polygon is smaller than the texture image itself, the texture map will be undersampled during rasterization. As a result, the texture mapping will be noisy and 'sparkly.' The purpose of MIP mapping is to remove this effect.

Multipass Texturing - The process of applying multiple textures to polygons in a model by performing multiple texturing passes (called blending) until the image is completed.

NURBS (Nonuniform Rational B-Spline) - A type of spline that can represent more complex shapes than a Bezier spline.

Occlusion - The effect of one object in 3-D space blocking another object from view.

Perspective Correct - Correctly applying textures onto a polygon that will appear in 3-D space, regardless of position or perspective. Without perspective correction, textures shimmer and dance about the polygon, causing visual deformation.

Phong Shading - A computation-intensive rendering technique that produces realistic highlights while smoothing edges between polygons. The Phong shading algorithm is best known for its ability to render precise, realistic specular highlights. During rendering, Phong shading achieves excellent realism by calculating the amount of light on the object at tiny points across the entire surface instead of at the vertices of the polygons. Each pixel representing the image is given its own color based on the lighting model applied at that point. Phong shading requires much more computation for the hardware than Gouraud shading.

Pixelation - The effect of individual pixels becoming visible. Generally this is the result of undersized texture maps being magnified when the view is set close to the polygon object.

Polygon - A 2-D figure (usually a triangle or rectangle) that is the building block of a 3-D screen object. It usually takes hundreds or thousands of polygons to form the skeleton of a 3-D object.

Polyline - A sequence of straight line segments where the end point of the first line segment is coincident with the start of the second segment, the endpoint of the second segment is coincident with the start of the third segment, and so on. A piecewise linear curve.

Polygon-based modeling - Representing 3-D objects as a set or mesh of polygons.

Primitives - Basic geometric shapes, such as cubes, cylinders, spheres, and donuts (or toruses).

Projection - The process of reducing three dimensions to two dimensions for display is called Projection. It is the mapping of the visible part of a 3-D object onto a 2-D screen.

Rasterization - Translating an image into pixels.

Ray Trace - A method of 3-D rendering in which light beams are followed individually from their source to the illuminated object and back to the camera. The resulting shading and illumination give a highly realistic appearance and are necessary for rendering true refraction, reflection, and transparency.

Reflective Mapping - A ray-tracing render effect that charts the image of surrounding objects onto the surface texture of the "reflective" object.

Rendering - Producing realistic imagery and animations from 3-D models.

Rendering Engine - "Rendering Engine" generically applies to the part of the graphics engine that draws 3-D primitives, usually triangles or other simple polygons. In most implementations, the rendering engine is responsible for interpolation of edges and "filling in" the triangle.

Scene graph - The scene graph contains nodes that describe objects and their properties. It contains hierarchically grouped geometry to provide an audio-visual representation of objects, as well as nodes that participate in the event generation and routing mechanism.

Shading - A coloring of an object using a technique such as Grouraud or Phong shading without texture mapping.

Software Rendering - A way of rendering 3-D scenes without the use of special 3-D hardware, relying on the CPU to do all the 3-D graphics processing work.

Specular Highlights - A lighting characteristic that determines how light should reflect off an object. Specular highlights are typically white and can move around an object based upon camera position.

Spline-based modeling - Representing 3-D objects as surfaces made up of mathematically derived curves (splines).

Tessellation - Tessellation is the process of subdividing a surface into smaller shapes. To describe object surface patterns, tessellation breaks down the surface of an object into manageable polygons. Triangles or quadrilaterals are two usually used polygons in drawing graphical objects because computer hardware can easy manipulate and calculate these two simple polygons.

Texel - Short for TEXture Element. A colored dot in a texture map; texture element. The smallest addressable unit in a texture map.

Texture - A 2-D bitmap pasted onto objects or polygons to add realism.

Texture Mapping - Improves visual detail by simulating the material properties of a surface. The simulation is achieved by mapping a stored bitmap onto the object. It is computationally intensive but can simulate represent real materials like bricks, wood, or steel.

Tweening - Also known as in-betweening; calculating the intermediate frames between two keyframes to simulate smooth motion.

Vertex - A corner of a polygon used to draw 3-D graphics on the screen. When an object moves, or when the user's viewpoint of the object changes, the computer must recalculate the new position in virtual space of each vertex. It does this with a process called matrix multiplication.

Volumetric Fogging - Atmospheric noise effect with depth, in addition to a defining plane. Frequently employed in 3-D scenes to reduce polygon count and avoid "pop-up" scenery.

Volumetric Lighting - Lighting effect defined by a particle system. The patterns formed by the particles give the impression of light shining through a dense medium such as fog or smoke.

Wireframe - All 3-D models are constructed from lines and vertices forming a dimensional image of the image. Then texture, shading, or motion can be applied.

Z-Buffer - A memory buffer that stores the depth of all pixels in a scene. The Z buffer determines which triangle of many overlapping is closest to the viewer. It is a common method of eliminating hidden surface removal (which the viewer should not see) in the rasterization process. An integer value is stored in RAM, the "z-buffer" for each point on the screen. Before each pixel is drawn, the existing z-buffer value is compared to the z (depth) of the object at that point; if the existing value is less, it is nearer, so the new pixel is discarded. However, if the new Z value is less, then it is written to replace the old value, and the corresponding pixel is updated.

Z-sorting - A process of removing hidden surfaces by sorting polygons in back-to-front order prior to rendering. Thus, when the polygons are rendered, the forward-most surfaces are rendered last. The rendering results are correct unless objects are close to or intersect each other. The advantage is not requiring memory for storing depth values. The disadvantage is the cost in more CPU cycles and limitations when objects penetrate each other.

Appendix B
Commercial Software Information

Alias/Wavefront
210 King Street East, Toronto
Ontario, Canada, M5A 1J7
(416) 362-9181

www.aliaswavefront.com

Maya	$7,500 to 16,000	PC and SGI

Arbor Image Corporation
5651 Plymouth Road
Ann Arbor, MI 48105
(734) 741-8700

www.arborimage.com

Draftsman PLUS 32	$339.00 to 2,895.00	PC

Autodesk
111 McInnis Parkway
San Rafael, CA 94903
(800) 964-6432
(415) 507-5000

www.autodesk.com

AutoCAD 2000	$3,750.00	PC
Architectural Desktop R2	$4,395.00	PC
AutoSketch	$79.95	PC
AutoCAD Overlay 2000	$1,395.00	PC
Planix Complete Home Design Suite	$36.29	PC

Bentley Systems

www.bentley.com

MicroStation/J	$2,009 (Gov.)	PC
MicroStation/J w/ Triforma	$3,196.35 (Gov.)	PC

Broderbund Software

(319) 395-0115
(800) 973-5111

www.broderbund.com

3D Home Design Suite Deluxe 3.0	$79.95	PC

Enhanced Graphics Solutions

4-th Likhachevskiy per. 17A
125438, Moscow, Russia
7 (096) 313-1853

www.egsolutions.com

E.G.S. Quick Vector	$300.00 to 1,500.00	PC

Eos Systems, Inc.

205-2034 West 12th Ave
Vancouver, Canada, V6J 2G2
(604) 732-6658

www.photomodeler.com

PhotoModeler	$ 795.00	PC

Global Majic Software

6767 Old Madison Pike Hwy
Suite 405
Huntsville, AL 35806
(877) 336-2542

www.3dlinx.com

3DlinX Standard Edition	$895.00	PC

GraphiSoft

235 Kansas Street, Suite 200
San Francisco, CA 94103
(415) 703-9777
(800) 344-3468

www.graphisoft.com

ArchiCAD 6.0	$4,295.00	PC

Hash Inc.

2800 East Evergreen Blvd.
Vancouver, WA 98661
(360) 750-0042

www.hash.com

Animation Master	$199.00	PC

IMSI

IMSI Corporate Headquarters
75 Rowland Way
Novato, CA 94945
(415) 878-4000

www.imsisoft.com

TurboCAD v6	$295.95	PC
TurboSketch	$69.95	PC
TurboCAD 3D Modeler	$99.95	PC
TurboCAD SolidModeler	$999.95	PC
Hijaak Pro	$295.95	PC
FloorPlan 3D Design Suite v5	$49.00	PC

Intergraph

Corporate Headquarters
Huntsville, AL 35894
(256) 730-2000
(800) 260-0246

www.intergraph.com

ModelView	$1,030	PC
SmartSketch	$199.00	PC

Kinetix

642 Harrison Street
San Francisco, CA 94107
(800) 879-4233

www.ktx.com

3D Studio MAX R3.0	$3,495.00	PC
3D Studio VIZ R3	$1,995.00	PC

Mak Technologies

380 Green Street
Cambridge, MA 02139
(617) 876-8085

www.mak.com

MakStealth	$~5,000	PC, SGI

MetaCreations

498 Seventh Avenue, 18th Floor
New York, NY 10018
(646) 485-9100
(800) 846-0111

www.metacreations.com

Carrara	$599.00	PC
Ray Dream Studio 5.5	$349.00	PC
Canoma	$469.00	PC

Microsoft

www.microsoft.com

Softimage 3D Extreme	$7,995 to 13,995	PC

MicroMouse Productions

847 Athol Street
Regina, Saskatchewan
Canada S4T 3B6
(306) 522-6077

www.quantumlynx.com/micromouse

AccuTrans 3D	$60.00	PC

Multigen

Corporate Headquarters
550 South Winchester Boulevard, Suite 500
San Jose, CA 95128
(408) 261-4100

www.multigen.com

Multigen Creator	$9,500	PC, SGI

NewTek

1200 SW Executive Drive
Topeka, KS 66615
(913) 228-8000
(800) TOASTER

www.newtek.com

LightWave 3D 5.6	$1,995	PC

NPSNET

Naval Post Graduate School

www.npsnet.nps.navy.mil/npsnet/bamboo.html

Bamboo	$Free	

Okino Computer Graphics
(888) 3D-OKINO
(905) 672-9328
www.okino.com

PolyTrans	$395.00	PC
	$495.00	Unix
PolyTrans OpenFlight plug-in	$245.00	PC
	$300.00	Unix

Palisades Research
869 Via De La Paz
Pacific Palisades, CA 90272
(310) 459-7528
www.aay.com

3D Builder Pro	$625.00	PC

Punch! Software, LLC
7900 NW 100th Street, Suite LL6
Kansas City, MO 64153
(816)891-0025
www.punchsoftware.com

Punch! Super Home Design Suite	$69	PC

Sense8
100 Shoreline Highway
Suite 282
Mill Valley, CA 94941
(800) EAI-7760
(415) 339-3220
www.sense8.com

WorldToolKit (developer license)	$5,000	PC
WorldToolKit (application license)	$6,000	PC

Softcover International Limited
320 Old Brompton Road
London SW5 9JH, England
00 44 20 7259-2100
www.softcover.com

Scan2CAD	$220.00	PC

SuperScape

3945 Freedom Circle
Suite 1050
Santa Clara, CA 95054
(408) 969-0500
(888) VWWW COM

www.superscape.com

SuperScape VRT	$3,400	PC
SuperScape VRT Plus	$6,500	PC
SuperScape 3D WebMaster	$495	PC
Viscape Universal	$Free	plug-in for web browser

Trix Systems, Inc.

68 Smith Street
Chelmsford, MA 01824
(800) 326-4443

www.trixsystems.com

TracTrix	$249.50 to 950.00	PC

Viewpoint Digital, Inc.

(801) 229-3212

www.viewpoint.com

Interchange 5.5	$495.00	PC

REPORT DOCUMENTATION PAGE

Form Approved
OMB No. 0704-0188

1. REPORT DATE *(DD-MM-YYYY)*
September 2001

2. REPORT TYPE
Final report

3. DATES COVERED *(From – To)*

4. TITLE AND SUBTITLE
Preliminary Investigation of the Visualization of Buildings in the OneSAF Environment

5a. CONTRACT NUMBER

5b. GRANT NUMBER

5c. PROGRAM ELEMENT NUMBER

6. AUTHOR(S)
Michael E. Pace, Warren R. Bennett

5d. PROJECT NUMBER

5e. TASK NUMBER

5f. WORK UNIT NUMBER

7. PERFORMING ORGANIZATION NAME(S) AND ADDRESS(ES)
U.S. Army Engineer Research and Development Center
Information Technology Laboratory
3909 Halls Ferry Road
Vicksburg, MS 39180-6199

8. PERFORMING ORGANIZATION REPORT NUMBER
ERDC/ITL SR-01-1

9. SPONSORING/MONITORING AGENCY NAME(S) AND ADDRESS(ES)
U.S. Army Corps of Engineers
Washington, DC 20314-1000

10. SPONSOR/MONITOR'S ACRONYM(S)

11. SPONSOR/MONITOR'S REPORT NUMBER(S)

12. DISTRIBUTION/AVAILABILITY STATEMENT
Approved for public release; distribution is unlimited.

13. SUPPLEMENTARY NOTES

14. ABSTRACT

The next generation of Models and Simulation (M&S) for use in determining future force design, developing doctrine, training military leaders, and performing acquisition for the Army After Next is currently under development. The Army M&S program is developing software called One-Semi Automated Force (OneSAF) to fulfill the objectives of M&S. Special Forces units train for terrorist scenarios such as capture and occupation of structures that may include innocent hostages. Given available information about the building at the time, a realistic simulation is desired in order to train to neutralize the threat. At present, OneSAF has a capability to perform military operations within buildings, but a rapid method to produce a building model for the synthetic environment does not exist.

In September 1999, ITL began an investigation of the visualization of buildings in the OneSAF environment. This report details the findings of this preliminary study.

(Continued)

15. SUBJECT TERMS

OneSAF	CADD	Modeling	Simulation
Building	MES	Multi-elevation	Visualization

16. SECURITY CLASSIFICATION OF: a. REPORT	b. ABSTRACT	c. THIS PAGE	17. LIMITATION OF ABSTRACT	18. NUMBER OF PAGES	19a. NAME OF RESPONSIBLE PERSON / 19b. TELEPHONE NUMBER *(include area code)*
UNCLASSIFIED	UNCLASSIFIED	UNCLASSIFIED		63	

14. ABSTRACT (Concluded).

This effort is an attempt to apply M&S tools to the specific problem of visualizing structure interiors for Special Forces operations. These operations are rapid response operations that must be conducted within hours of the initial invasion of the structure. A target deadline of 3 days has been given as the metric for performing the modeling and training for the operation.

This report is organised into chapters that address the state of modeling and capabilities that presently exist pertaining to buildings. A discussion of general modeling processes to proceed from a paper format to the format required for the OneSAF program are presented. Finally, various software packages, file formats, and programming controls and Application Programmer Interfaces are discussed that might shorten the modeling time required.